

SYSTEM AND METHOD FOR SOFTWARE SELLING

Background Of The Invention

5

This invention generally relates to methods and systems for selling software applications. More specifically, the invention relates to methods and systems for encouraging users to purchase software applications after being provided with trial or demonstration versions or samples of the applications.

10

Companies often make programs for evaluation purposes. Such software applications may be run on a computer, kiosk, web-TV, or other device. Often, such applications display text and images in windows on monitors and provide graphical user interfaces with which users interact with the applications. Typical applications include text and image editors, mail systems, web browsers, financial programs, games, drawing programs, music players, and various office application software. The software vendor may send the programs to consumers for free. Consumers can download programs from the Internet or load them on to their computers from a CD, and the consumers can use the programs during a certain period of time.

20

Most of the time, the programs have expiration dates beyond which consumers cannot use the programs. For example, a user may be able to download a program such as a text editor and use it for one month until the program expires. Close to the end of the expiration period, the program starts to send a warning message to the user, informing the user that he needs to buy the licensed version of the program, otherwise the program will expire. If the user does not buy the program, the demo program will stop working after the evaluation period. Also, the user can buy a software key (typically a code) that will remove the expiration date and allow the program to continue to work.

25

This approach is often problematic. When the program stops working, the user often decides to find another company that makes a similar program with similar functionality. Often, the person goes back to the vender's web site, again downloads
5 the demo version of the program, and uses it for another month until it too expires, without ever buying the program.

A current challenge to manufacturers is to determine ways to encourage users to buy the trial products. Manufacturers need methods that are more effective than simply
10 suddenly stopping the use of the product.

Summary Of The Invention

An object of this invention is to improve systems and methods for selling software.
15

Another object of the present invention is to encourage consumers to purchase software applications after being given a free trial or demonstration version or sample of the application.

20 A further object of the invention is to control the appearance and performance of a software application.

Another object of this invention is to progressively degrade a trial or demonstration version of a software program in a manner that still permits use of the trial or
25 demonstration version while encouraging the user to purchase the software program.

These and other objects are attained with methods and system to progressively degrade a trial version of the product in various ways so as to permit the user to continue to use the product until the user becomes reliant on the product and becomes

so frustrated that the user purchases the product. This approach, it may be noted, is also useful for discouraging illegal use of software without proper licenses.

5 The trial or demonstration version of the software program may degrade in any one or more of a number of ways. For instance, the brightness may fade, the application window may shrink, the font may degrade, features may drop out, the display quality may degrade, the sound quality may diminish, the processing speed may decrease, or the network bandwidth may decrease.

10 Further benefits and advantages of the invention will become apparent from a consideration of the following detailed description, given with reference to the accompanying drawings, which specify and show preferred embodiments of the invention.

15 Brief Description Of The Drawings

Figure 1 is an illustrative example of this invention.

Figure 2 describes how components of a software program may change.

20

Figure 3 shows how the intensity of a program can change.

Figure 4 is a flow chart of a preferred embodiment of this invention.

25

Detailed Description Of The Preferred Embodiments

As discussed above, companies often make software programs available for evaluation purposes. Such software applications may be run on a computer, kiosk,

web-TV, or other device. Often, such applications display text and images in windows on monitors and provide graphical user interfaces with which users interact with the applications. Typical applications include text and image editors, mail systems, web browsers, financial programs, games, drawing programs, music players,
 5 and various office application software.

In accordance with the present invention, a trial or demonstration version or sample of the program is provided with means to diminish or decrease the quality of the program over time. The software program may be gradually degraded in any one or
 10 more of a number of ways. For example, 1) The brightness fades so that it is harder for the user to use the software tool as time progresses. This is the user's inducement to pay for the license, registration code, or actual software. 2) The application window shrinks through time. 3) The font degrades. 4) Features gradually drop out. 5) Display quality (spatial resolution, color resolution, display refresh rate) degrades.
 15 6) Sound quality degrades. 7) Processing speed degrades. 8) Network bandwidth degrades. 9) Unpleasant visual and auditory stimulus is produced. For examples, watermarks may be superimposed and become progressively more opaque; blinking, interfering photographs may be shown. 10) Information degrades (for text, some letters are progressively omitted, words are progressively omitted, some sentences are
 20 progressively omitted, some document sections are progressively omitted. For application with sound, sound sections are progressively omitted. Random omissions may also be performed.

For example, after thirty days, the application may be 90% as bright as it was
 25 originally, and after sixty days the application may be 80% as bright. This is the user's inducement to pay for the license, registration code, and/or fully-functioning software. Similarly, the application window may shrink in size through time. The font may degrade through time. For example, pieces of letters may progressively or randomly fade, be omitted, be made more complex, or made more obscure.

Various program features and options may gradually become unavailable for use by the user. For example, a drawing package that allows a user to draw lines, circles, polygons, stars, and spirals, may one day make the "circle drawing" option
5 unavailable and on another day make the "polygon drawing" option unavailable. Display quality (such as spatial resolution, color resolution, and refresh rate) may
10 degrade through time. Sound quality may degrade through time and/or the volume may diminish through time. Processing speed may degrade so that, for example, a task that takes a certain amount of time one day takes may take more time the next day.

Network bandwidth speeds may degrade, which would be particularly relevant to software that requires use of a network like the Internet. Unpleasant visual, auditory, olfactory, and tactile stimulus may be produced. For example, watermarks may be
15 superimposed on an application's display window and become progressively more opaque.

The system may make graphics blink unpleasantly or present unpleasant photographs, sounds, and odors. Information may degrade. For example, for text, some letters are
20 progressively omitted, words are progressively omitted. For applications with sound, segments of sound are progressively omitted. All of these examples of degradation can happen in a random manner as well as a gradual, progressive manner. The purpose of the degradation is to induce users to buy the product software.

25 Figure 1 illustrates an example of this invention. Inside of the computer system 100 there is the program 106. This is a software product that the user 107 received as a demo version. This program was received by the user from the store 101. A "store" may be physical or electronic (e.g. a web site). The program received by the user has a time constraints 104 on how long the program will exist in its original form. For

Figure 2 describes how the components of a program may change. 200 shows all the components of the program that are changing. The change of “color” includes intensity, saturation, hue, resolution, etc. Intensity 206 of content 212 becomes less intense, less bright, content is less visible. 202 show the text structure changing. For example, fonts begin to change, they become smaller, less readable, fonts shape become curved. The focus diminishes to make the text less readable. 203 shows the contour, edges of the different shapes on the screen. This contour becomes less visible and less usable to make the program less useful. 204 shows the speed of the performance of the different components of the software program. Certain functions are not set.

YOR920000475US1

managed by the timer 230. Timer controls the transformator 210 that is 103 on Figure 1. The transformator 210 changes the features described above (color, texture, speed etc.) depending on the timer 230. Every component depends on the timer and changes as the time progresses.

5

Figure 3 shows an example of how the intensity can be changed. In this example, the software program presents two pictures with different colors. The Figure shows a broken line, marked by "part 1", as one color or intensity, and a solid line, "part 2", as a different color or intensity. The program contains a table that shows the dependencies on timer as to how to change the intensity. As the time progresses, the intensities of these two lines change. For example, at sometime the intensity of part 2 (Figure 3) becomes brighter then the intensity of part 1, which makes it difficult for the user to observe the screen.

Figure 4 shows the flow chart of the business model of how to use this approach. In this model, at 400, a user receives the demo program. He can use this program for some time period 401. After this evaluation time, the user is required to purchase this program. If the user does not buy the license, some components start to change and the program ages and degrades 402. The degradation or lack of degradation may be controlled by the degradation server provider 109, particularly if the system 100 is attached to a network 106 or if a portion of program resides on remote computer 108. The color, text, speed degrade so it is difficult to use this program successfully. After some time period, use of the program is nearly impossible. This approach encourages the user to buy the program, instead of using the old program 403.

25

As a result, in order to use the program, the user is encouraged to purchase the license to remove the degraded features.

The present invention has a number of important advantages. For instance, the customer who is using the full program easily becomes used to the program, its functionality, program performance, etc. After a certain period of time, during which the user becomes comfortable with the program, the program slowly degrades. The user becomes frustrated with the degraded functionality and performance and, in frustration, the chances increase that the user will purchase the full program. The customer will get used to the program because he/she is using it every day. And when he observes the degradation, he will have a great incentive to buy the product.

- 10 While it is apparent that the invention herein disclosed is well calculated to fulfill the objects stated above, it will be appreciated that numerous modifications and embodiments may be devised by those skilled in the art, and it is intended that the appended claims cover all such modifications and embodiments as fall within the true spirit and scope of the present invention.

15